User guide:

Introduction:

This game is called *Maze*, made by Peter, Emily, and Luke, where player should move the character on the screen to get out of the maze. Our program used “if()” statements, and “for()” loops to run the code. We had 10 variables in total and 20+ “if()” statements. We have a one minute timer. When timer counts down to 0, the programme will end, and player will fail the game; on the other hand, if you reach the end square in time, you will win the game and the screen will show an “complete”. Enjoy.

How to Play:

1. Press “w, s, a, d” to move character
2. “w” to move back, “s” to move front, “a” to move left, “d” to move right
3. Avoid red enemies!
4. Don’t touch walls and obstacles
5. Get to the end square before the timer counts down to 0

How the code works:

For the movement of the main character we used keyPressed to indicate if any key was pressed and then move the character somewhere based on the key. In this part we used the variables with which we set the position of the playable character

The movement of enemies works by adding 1 pixel to their x coordinate in a certain direction, if the x coordinate reaches a certain point, the amount added is multiplied by -1

Limited vision works by separating the entire map into 16 squares and drawing every square apart of the one in which the character is currently located in. As we draw each square manually, we created a variable that will make it faster to write

Resetting the main character works by comparing the values of the colour of pixels just outside of 8 points around the circle. The colour of these points is then compared, and if any of those are not equal, the character resets to the beginning of the map

The timer works by setting a certain amount and subtracting 1 on each frame. Since we are using 60 frames per second we multiplied the timer by 60 and ended with 3600. When we are printing the value of the time, we divide the timer by 60 to get the value in seconds. Once it runs out it prints FAIL and reveals the map.

The end zone is waiting until the coordinates of the main character reach it, and it then prints COMPLETE and reveals the map

If the game ends, either by completing it or failing it will reveal the map and stop the motion of all characters. This works by having a variable fail which is set to true once we want the game to end. All parts we want stopped have an if statement in front of them